

PATENT

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Appl. No.: 10/518,930 Confirmation No.: 7618  
Appellant(s): Koistinen et al.  
Filed: January 13, 2005  
Art Unit: 2446  
Examiner: Shaq Taha  
Title: LOAD BALANCING DEVICES AND METHOD THEREFOR

Docket No.: 042933/373913  
Customer No.: 00826

Mail Stop Appeal Brief-Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF UNDER 37 CFR § 41.37**

This Appeal Brief is filed pursuant to the "Notice of Appeal to the Board of Patent Appeals and Interferences" and "Pre-Appeal Brief Request for Review" filed July 9, 2009 and the "Notice of Panel Decision from Pre-Appeal Brief Review," mailed September 18, 2009.

**Table of Contents:**

1. Real Party in Interest, begins on Page 2;
2. Related Appeals and Interferences, begins on Page 3;
3. Status of Claims, begins on page 4;
4. Status of Amendments, begins on page 5;
5. Summary of Claimed Subject Matter, begins on page 6;
6. Grounds of Rejection to be reviewed on Appeal, begins on page 11;
7. Argument, begins on page 12;
8. Claims Appendix, begins on page 19;
9. Evidence Appendix, begins on page 25; and
10. Related Proceedings Appendix, begins on page 26.

In re: Koistinen et al.  
Appl. No.: 10/518,930  
Filing Date: 1/13/2005

1. ***Real Party in Interest.***

The real party in interest in this appeal is Nokia Corporation, the assignee of the above-referenced patent application.

In re: Koistinen et al.  
Appl. No.: 10/518,930  
Filing Date: 1/13/2005

2. ***Related Appeals and Interferences.***

There are no related appeals and/or interferences involving this application or its subject matter.

In re: Koistinen et al.  
Appl. No.: 10/518,930  
Filing Date: 1/13/2005

3. ***Status of Claims.***

The present application includes pending Claims 26-37 and 41-56, all of which stand rejected and are the subject of the present appeal.

In re: Koistinen et al.  
Appl. No.: 10/518,930  
Filing Date: 1/13/2005

4. ***Status of Amendments.***

There are no un-entered amendments in this application.

5. ***Summary of Claimed Subject Matter.***

Independent claim 26 recites a method including obtaining a current connection state (*See Pat. Appl. at p.7, lines 31-34*) as well as a current load state (*See Id. at p. 8, lines 1-5*) of each of a plurality of processors (*See Id. FIG. 1, reference no. 11*) configured to perform communication in a packet switched connection (*See Id. at p.10, lines 17-25*). The method further including selecting on a per received packet basis, by a load balancer (*See Id. FIG. 1, reference no. 12*) configured to distribute load to the processors, a processor having a lowest load in such a manner that a respective next received packet is distributed to the processor irrespective of a specific connection to which the next received packet belongs. *See Id. at p.7, lines 26-30 and p.10, lines 26-31*. The method further reciting maintaining information about the load state of each processor so that selecting is performed by selecting one of the processors to serve and process a respective received packet based on the load state. *See Id. at p.8, lines 1-5*. The method also including informing the current connection state to respective processors including inserting data indicating the current connection state into a packet to be distributed. *See Id. at p.8, lines 12-18*.

Claim 27, depending from claim 26, further recites that the data storage (*See Id. FIG. 1, reference no. 14*) is accessed by the load balancer. *See Id. at p.3, lines 8-9*.

Claim 28, depending from claim 26, further recites that the data storage is accessed by the processors. *See Id. at p.4, lines 26-29*.

Claim 29, depending from claim 26, further recites that information about the load state is maintained as a Boolean state. *See Id. at p.3, lines 8-11 and p.8, lines 1-4*.

Claim 30, depending from claim 26, further recites that a processor is selected in a round-robin fashion. *See Id. at p.8, lines 5-7*.

Claim 31, depending from claim 26, further recites that a supported service profile for each processor is maintained. *See Id. at p.5, lines 9-13 and p.8, lines 7-10*.

Claim 32, depending from claim 31, further recites that the supported service profile is used as an additional selection criteria. *See Id. at p.5, lines 9-13 and p.8, lines 7-10*.

Claim 33, depending from claim 26, further recites that the load balancer is configured to obtain a load state from each processor upon a hardware based mechanism. *See Id.* at p.8, lines 12-15.

Claim 34, depending from claim 26, further recites that the load balancer is configured to obtain a load state from each processor upon a packet based switching mechanism. *See Id.* at p.8, lines 12-18.

Claim 35, depending from claim 34, further recites that a load state of a processor is inserted into a received packet processed by the processor. *See Id.* at p.8, lines 12-18.

Claim 36, depending from claim 35, further recites that a packet returned by a processor is interpreted as a flag for a free resource. *See Id.* at p.8, lines 12-18.

Claim 37, depending from claim 26, further recites that excess traffic is redirected to another load balancer, the excess traffic being defined based upon the number of active processors. *See Id.* from p.9, line 31 to p.10, line 2.

Independent claim 41 recites an apparatus including selection circuitry configured to select, on a per received packet basis, one of a plurality of processors (*See Id.* FIG. 1, reference no. 11) configured to perform communication in a packet switched connection (*See Id.* at p.10, lines 17-25) on the basis of a stored load state of the selected processor in such a manner that a respective next received packet is distributed to the selected processor with a lowest load among the processors, irrespective of a specific connection to which the next received packet belongs. *See Id.* at p.8, lines 1-7. The apparatus further including connection state informing circuitry configured to inform the current connection state to respective processors by inserting data indicating the current connection state into a packet to be distributed. *See Id.* at p. 8, lines 28-34.

Claim 42, depending from claim 41, further recites that a load state of a processor is contained in a table. *See Id.* at p. 9, lines 1-4.

Claim 43, depending from claim 41, further recites that a load state of a processor is expressed as a Boolean value. *See Id.* at p.8, lines 1-3 and p.9, lines 1-4.

Claim 44, depending from claim 41, further recites that a load state of a processor is expressed as a value that corresponds to the percentage of load. *See Id.* at p. 9, lines 1-4.

Claim 45, depending from claim 41, further recites that the selection circuitry is configured such that a processor is selected also on the basis of a parameter indicating the service profile supported by a respective processor. *See Id.* at p.5, lines 9-13 and p.8, lines 7-10.

Claim 46, depending from claim 45, further recites that the parameter is contained in a table. *See Id.* at p.5, lines 9-13.

Claim 47, depending from claim 41, further recites that the apparatus also includes data insertion circuitry configured to insert a communication connection state into a received packet to be routed. *See Id.* at p.5, lines 15-17.

Claim 48, depending from claim 41, further recites that the processors include multicore digital signal processing elements having shared data storage for all cores, whereby the device includes a first level of load balancing configured to select a digital signal processing means and a second level of load balancing configured to select a single core. *See Id.* at p. 5, lines 19-24.

Claim 49, depending from claim 41, further recites that the apparatus also includes a switch configured to redirect excess traffic to another apparatus where the excess traffic is defined based upon the number of active processors. *See Id.* at p. 5, lines 25-29.

Independent claim 50 recites a system including an obtaining unit configured to obtain a current connection state as well as a current load state of each of a plurality of processors (*See Id.* FIG. 1, reference no. 11) configured to perform communication in a packet switched connection. *See Id.* at p.10, lines 17-25. The system further includes a selector configured to select on a per received packet basis one of the processors, by a load balancer (*See Id.* FIG. 1, reference no. 12) configured to distribute load to the processors in such a manner that a respective next received packet is distributed to the selected processor having a lowest load irrespective of a specific connection to which this next received packet belongs. *See Id.* at p.10, lines 25-30. The system also includes a maintenance unit configured to maintain information about the load state of each processor so that the selecting includes selecting one of the processors to serve and process a respective received packet based on the load states. *See Id.* at p.8, lines 1-5. The system further includes an informing unit configured to inform the current connection state to respective processors including inserting data indicating the current connection state into a packet to be distributed. *See Id.* at p.8, lines 12-18

Independent claim 51 recites a computer program embodied on a computer readable medium, the computer readable medium storing code including computer executable instructions configured to perform a method that includes obtaining a current connection state as well as a current load state of each of a plurality of processors (*See Id.* FIG. 1, reference no. 11) configured to perform communication in a packet switched connection. *See Id.* at p.10, lines 17-25. The method further includes selecting on a per received packet basis, by a load balancer (*See Id.* FIG. 1, reference no. 12) configured to distribute load to the processors, one of the processors in such a manner that a respective next received packet is distributed to the selected one of the processors having a lowest load irrespective of a specific connection to which a respective received packet belongs. *See Id.* at p.7, lines 26-30 and p.10, lines 26-31. The method also includes maintaining information about the load state of each processor so that the selecting includes selecting one of the processors to serve and process a respective received packet based on the load state. *See Id.* at p.8, lines 1-5. The method further includes informing the current connection state to the respective processors including inserting data indicating the current connection state into a packet to be distributed. *See Id.* at p.8, lines 12-18.

Independent claim 52 recites a system that includes a plurality of processors (*See Id.* FIG. 1, reference no. 11) configured to perform communications in a packet switched connection and at least one load balancer (*See Id.* FIG. 1, reference no. 12) configured to distribute load to the processors. *See Id.* at p.10, lines 17-25. The load balancer is configured to obtain a stored current connection state and a current load state of each of the processors, maintain information about the load state of each of the processors, and select a processor in such a manner that a respective next received packet is distributed to the processor having the lowest load irrespective of a specific connection to which a respective received packet belongs. *See Id.* at p.7, lines 26-30 and p.10, lines 26-31. The load balancer is further configured to inform the current connection state to respective processors including inserting data indicating the current connection state into a packet to be distributed. *See Id.* at p.8, lines 12-18.

Independent claim 53 recites an apparatus including a load balancer (*See Id.* FIG. 1, reference no. 12) where the load balancer is configured to obtain a current connection state and a current load state of each of a plurality of processors (*See Id.* FIG. 1, reference no. 11) and

maintain information about the load state of each of the processors. *See Id.* at p.7, line 31 to p.8, line 5. The load balancer is further configured to select, on a per received packet basis, a processor having a lowest load in such a manner that a respective next received packet is distributed to the processor irrespective of a specific connection to which this next respective received packet belongs. *See Id.* at p.7, lines 26-30 and p.10, lines 26-31.

Independent claim 54 recites an apparatus including means for maintaining a load state of each of a multitude of processors (*See Id.* FIG. 1, reference no. 11) performing a packet switched communication connection. *See Id.* at p.10, lines 17-25. The apparatus also includes selecting means for selecting, on a per received packet basis, one of the processors on the basis of its load state in such a manner that a respective next received packet is distributed to a processor having a lowest load irrespective of a specific connection to which a respective packet belongs. *See Id.* at p.7, lines 26-30 and p.10, lines 26-31. The apparatus further includes informing means for informing the current connection state to respective processors by inserting data indicating the current connection state into a packet to be distributed. *See Id.* at p.8, lines 12-18.

Claim 55, depending from claim 54, further includes means for inserting a communication connection state into a packet to be routed. *See Id.* at p.8, lines 12-18.

Claim 56, depending from claim 54, further includes means for redirecting excess traffic to another device, where the excess traffic is defined based upon the number of active processors. *See Id.* from p.9, line 31 to p.10, line 2.

In re: Koistinen et al.  
Appl. No.: 10/518,930  
Filing Date: 1/13/2005

6. ***Grounds of Rejection to be reviewed on Appeal.***

The following grounds of rejection are appealed:

(A) Claims 26-37, 41-47, and 49-56 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication No. 2003/0187914 to Kaniyar et al.

(hereinafter “Kaniyar”) in view of U.S. Patent No. 6,976,085 to Aviani et al. (hereinafter “Aviani”).

(B) Claim 48 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Kaniyar in view of Aviani, and in further view of U.S. Patent Application Publication No. 2002/0059502 to Reimer et al. (hereinafter “Reimer”).

7. *Argument.*

(A) The Rejections of Claims 26-37, 41-47, and 49-56

Independent claims 26, 41, and 50-54 are argued together below. Additionally, dependent claims 31, 33, 35, and 45 are separately argued. Although not all of the dependent claims are separately argued, it is believed that each of the dependent claims is separately patentable.

The Final Office Action of May 4, 2009 has taken the position that Kaniyar discloses all of the elements of the claims, with the exception of informing the current connection state to respective processors comprising inserting data indicating the current connection state into a packet to be distributed. The Office Action then cites Aviani as allegedly curing this deficiency in Kaniyar. Applicant traverses this rejection as the combination of Kaniyar and Aviani does not disclose or suggest all of the limitations of the claims.

Kaniyar discloses a method for implementing symmetrical multiprocessing in a multiprocessor system and increasing performance of the multiprocessor system. More particularly, Kaniyar is directed to symmetrically partitioning I/O tasks for network connections across processors in the multiprocessor system so that each connection state lives on a single processor for its lifetime. As a result, Kaniyar ensures that I/O tasks associated with a particular connection are processed by the same processor.

Aviani discloses a system that operates in a data communications device such as a switch or a router to provide a technique for inserting data into packets associated with a communication session between first and second computerized devices. The technique comprises receiving a first packet containing data being propagated from the first computerized device to the second computerized device in the communications session and inserting a first amount of extra data into the first packet to alter the size of the first packet, and then forwarding the first packet including the first amount of extra data to the second computerized device. By monitoring and adjusting sequence and acknowledgement information from within the data communications device, data can be inserted into packets without disrupting connection state information maintained by and expected by each computerized device.

Appellants respectfully submit that the combination of Kaniyar and Aviani fails to disclose or suggest all of the elements of the present claims. For example, the combination of Kaniyar and Aviani does not disclose or suggest “selecting on a per received packet basis, by a load balancer configured to distribute load to said processors, a processor having a lowest load in such a manner that a respective next received packet is distributed to the processor irrespective of a specific connection to which this next received packet belongs,” as recited in claim 26, and the similar elements recited in claims 41 and 50-54. Thus, according to the claimed invention, a received data packet is distributed to one of the processors irrespective of a specific connection to which the received packet belongs.

Kaniyar and Aviani, on the other hand, fail to disclose or suggest that a processor having a lowest load is selected, on a per received packet basis, in such a manner that a respective next received packet is distributed to the processor irrespective of a specific connection to which this next received packet belongs. The Final Office Action took the position that paragraph 0031 of Kaniyar discloses this limitation of the claims (see Final Office Action, page 3). Paragraph 0031 of Kaniyar discloses that the least busy processor in the system can be selected as a “scheduling processor.” Therefore, this section of Kaniyar is directed to choosing a “scheduling processor,” which is the processor that determines which processor in the system will process the packet. Accordingly, in Kaniyar, one of the processors is selected as the load balancing processor or “scheduling processor.”

According to the claimed invention, on the other hand, a separate load balancing unit is provided to distribute the load. Furthermore, the claimed invention, the load balancing unit selects a processor to process a received packet based on a lowest load and irrespective of a specific connection to which the received packet belongs. In contrast, according to Kaniyar, the “scheduling processor” selects the processor that has previously processed packets from the same network connection. In other words, Kaniyar teaches that packets received from the same network connection are scheduled for processing by the same processor, so that each connection state lives on a single processor for its lifetime (Kaniyar, paragraphs 0007, 0008, 0033). Thus, the goal of Kaniyar is to systematically partition data streams for connections across processors

to enable a connection state to live on a single processor for the lifetime of the connection which, in turn, enhances performance of the multiprocessor system (Kaniyar, paragraph 0033).

In view of the above, Appellants respectfully submit that Kaniyar fails to disclose or suggest “selecting on a per received packet basis... a processor having a lowest load in such a manner that a respective next received packet is distributed to the processor **irrespective of a specific connection to which this next received packet belongs**,” as recited in claim 26, and similarly recited in claims 41 and 50-54. Rather, in direct contrast to what is recited in the present claims, Kaniyar discloses that packets received from the same network connection are always scheduled for processing by the same processor, so that each connection state lives on a single processor for its lifetime.

Additionally, Kaniyar recites “The mapping algorithm...ensures data packets received from the same network connection are routinely scheduled for processing by the same selected processor in the multiprocessor system.” Paragraph 8, page 2. Therefore, Kaniyar would not benefit from “informing the current connection state to respective processors comprising inserting data indicating the current connection state into a packet to be distributed” as recited in Claim 26 because the same connection is handled by the same processor which can store the connection state. The recited element of claim 26 would be duplicative and counter-intuitive to the teachings of Kaniyar. Similar elements are recited in claims 41 and 50-54.

Furthermore, Aviani does not cure these deficiencies in Kaniyar. Aviani, as discussed above, merely discloses inserting connection information into data packets. Aviani, like Kaniyar, fails to disclose or suggest that a processor having a lowest load is selected, on a per received packet basis, in such a manner that a respective next received packet is distributed to the processor irrespective of a specific connection to which this next received packet belongs. Therefore, Appellants respectfully assert that the combination of Kaniyar and Aviani does not disclose or suggest “selecting on a per received packet basis, by a load balancer configured to distribute load to said processors, a processor having a lowest load in such a manner that a respective next received packet is distributed to the processor irrespective of a specific connection to which this next received packet belongs,” as recited in claim 26, and the similar

limitations recited in claims 41 and 50-54. In view of the above, it is respectfully requested that the rejection of claim 26, 41, and 50-54 be overturned.

While the traversal of the rejections of the independent claims serves to traverse the rejections of the dependent claims, several dependent claims are herein argued separately. As each independent claim is believed to be in condition for allowance, all dependent claims are believed to be in similar condition; however Appellant noted with particularity the following claims and rejections.

Claim 31, which depends from claim 26 and includes all of the elements therefrom, further recites that a supported service profile is maintained for each processor. The Office Action indicates that Kaniyar teaches this element reciting “The NIC, which maintains a processor queue for each processor in the system, then queues the packet descriptor to the appropriate processor queue based on the hash value, (Kaniyar et al., Paragraph 10, Page 2).” Appellant respectfully disagrees as Kaniyar clearly teaches that a “processor queue” is maintained versus the present claim of a “supported service profile”, an example of which is given in the specification as GSM codecs (*See Application*, page 8, lines 7-10). Appellant asserts that a “processor queue” is substantially different from a “supported service profile” and therefore, the rejection of claim 31 is overcome.

Claim 33, which depends from claim 26 and includes all of the elements therefrom, further recites that the “load balancer is configured to obtain a load state from each processor upon a hardware based mechanism.” The Office Action cites Kaniyar, paragraph 26, page 3 as teaching this element. Appellant respectfully disagrees as the cited language recites “a suitable hardware structure for achieving scalability beyond a single processor is a ‘symmetric multiprocessor’ (SMP) system.” Kaniyar does not teach or suggest that the load state of each processor of the “multiprocessor system” is obtained upon a hardware based mechanism, nor is any disclosure cited that would allude to such teachings. Therefore, the rejection of claim 33 is overcome.

Claim 35, which depends from claim 34, which depends from claim 26, recites all of the elements of claims 26 and 34, and further recites that “a load state of a processor is inserted into

a received packet processed by said processor.” The Office Action admits that Kaniyar fails to teach that the load state is inserted into a packet processed by said processor, and cites Aviani to correct this deficiency. Appellant asserts that Avani is improperly characterized and the combination of references fails to teach the aforementioned element of claim 35. The rejection of claim 35, on page 7 of the Final Office Action of May 4, 2009 does not allude to either Avani or Kaniyar teaching that the “load state of the processor is inserted into a packet processed by the processor.” The rejection fails to cite any disclosure from either reference that teaches or suggests any state (connection or load) information is inserted into a packet processed by the processor, much less a load state. Therefore the Appellant believes that the rejection is improper and poorly articulated. Claim 47 recites a similar element and the identical rejection is used in the Office Action on pages 10 and 11. Appellant respectfully asserts that the rejections of claims 35 and 47 are overcome.

Claim 45, depending from claim 41 and including all of the elements therefrom, further recites “an apparatus wherein said selection circuitry is configured such that a processor is selected also on the basis of a parameter indicating the service profile supported by a respective processor.” The Office Action cites Kaniyar, paragraph 40, page 5 as teaching this element. Kaniyar recites “If the data packet is not of the type that should be scaled, in step 508, the selected processor is chosen based on other load balancing criteria[.]” Appellant believes that the Examiner is using impermissible hindsight as “other load balancing criteria” cannot be reasonably interpreted to include a “supported service profile” without the advantage of the disclosure of the present application. Therefore, the rejection of claim 45 is overcome.

Beyond the separately argued dependent claims, claims 27-37, 42-49, 55, and 56 are dependent upon claims 26, 41, and 54 respectively. As such, claims 27-37, 42-49, 55, and 56 should be allowed for at least their dependence upon claims 26, 41, and 54, and for the specific elements recited therein.

#### (B) The Rejection of Claim 48

The Final Office Action of May 4, 2009 took the position that Kaniyar and Aviani disclose all of the limitations of claim 48, with the exception of processors being comprised of

In re: Koistinen et al.  
Appl. No.: 10/518,930  
Filing Date: 1/13/2005

multicore digital signal processing elements having a shared data storage for all cores, whereby said device comprises a first level of load balancing configured to select a digital signal processing means and a second level of load balancing configured to select a single core. The Office Action then cited Reimer to allegedly cure this deficiency of Kaniyar and Aviani.

Kaniyar and Aviani are discussed above with regard to independent claim 41, from which claim 48 depends. Though the rejection of claim 48 has been traversed based on the traversal of the rejection of claim 41, Appellants will take this opportunity to further address the separate rejection posed by the Office Action.

Reimer discloses a multi-core digital signal processor having a shared program memory with conditional write protection. The digital signal processor includes a shared program memory, an emulation logic module, and multiple processor cores each coupled to the shared program memory by corresponding instruction buses. The emulation logic module determines whether the processors are operating in a normal mode or an emulation mode. In the emulation mode, the emulation logic can alter the states of various processor hardware and the contents of various registers and memory. The instruction buses each include a read/write signal that, while their corresponding processor cores are in the normal mode, is maintained in a read state. When the processor cores are in the emulation mode, the processor cores are allowed to determine the state of the instruction bus read/write signals. Each instruction bus read/write signal is generated by a logic gate that prevents the core from affecting the read/write signal value in normal mode, but allows the processor core to determine the read/write signal value in emulation mode.

Claim 48 is dependent upon claim 41 and inherits all of the limitations thereof. As discussed above, the combination of Kaniyar and Aviani fails to disclose or suggest all of the elements of claim 41. Further, Reimer fails to cure the deficiencies of Kaniyar. And Aviani, as Reimer also fails to disclose or suggest “selection circuitry configured to select, on a per received packet basis, one of a plurality of processors configured to perform communication in a packet switched connection on the basis of a stored load state of the selected processor in such a manner that a respective next received packet is distributed to the selected processor with a lowest load among said processors irrespective of a specific connection to which this next received packet

In re: Koistinen et al.  
Appl. No.: 10/518,930  
Filing Date: 1/13/2005

belongs," as recited in claim 41. Thus, the combination of Kaniyar, Aviani, and Reimer fails to disclose or suggest all of the elements of claim 48.

For at least the reasons discussed above, Appellants respectfully submit that the cited prior art fails to disclose or suggest all of the elements of the claimed invention. These distinctions are more than sufficient to render the claimed invention unanticipated and unobvious. It is therefore respectfully requested that all of claims 26-37 and 41-56 be allowed, and this application passed to issue.

8. ***Claims Appendix.***

The claims currently on appeal are as follows:

Claims 1-25. (Cancelled).

26. (Previously Presented) A method, comprising:

obtaining a current connection state as well as a current load state of each of a plurality of processors configured to perform communication in a packet switched connection;

selecting on a per received packet basis, by a load balancer configured to distribute load to said processors, a processor having a lowest load in such a manner that a respective next received packet is distributed to the processor irrespective of a specific connection to which this next received packet belongs;

maintaining information about the load state of each processor so that said selecting is performed by selecting one of said processors to serve and process a respective received packet based on the load state; and

informing the current connection state to respective processors comprising inserting data indicating the current connection state into a packet to be distributed.

27. (Previously Presented) A method according to claim 26, wherein said data storage is accessed by said load balancer.

28. (Previously Presented) A method according to claim 26, wherein said data storage is accessed by said processors.

29. (Previously Presented) A method according to claim 26, wherein said information about the load state is maintained as a Boolean state.

30. (Previously Presented) A method according to claim 26, wherein a processor is selected in a round-robin fashion.

31. (Previously Presented) A method according to claim 26, wherein a supported service profile for each processor is maintained.

32. (Previously Presented) A method according to claim 31, wherein said supported service profile is used as additional selection criteria.

33. (Previously Presented) A method according to claim 26, wherein said load balancer is configured to obtain a load state from each processor upon a hardware based mechanism.

34. (Previously Presented) A method according to claim 26, wherein said load balancer is configured to obtain a load state from each processor upon a packet based mechanism.

35. (Previously Presented) A method according to claim 34, wherein a load state of a processor is inserted into a received packet processed by said processor.

36. (Previously Presented) A method according to claim 34, wherein a packet returned by a processor is interpreted as a flag for a free resource.

37. (Previously Presented) A method according to claim 26, wherein excess traffic is redirected to another load balancer, said excess traffic being defined upon the number of active processors.

Claims 38-40. (Cancelled)

41. (Previously Presented) An apparatus, comprising:  
selection circuitry configured to select, on a per received packet basis, one of a plurality of processors configured to perform communication in a packet switched connection on the basis of a stored load state of the selected processor in such a manner that a respective next received packet is distributed to the selected processor with a lowest load among said processors irrespective of a specific connection to which this next received packet belongs; and

connection state informing circuitry configured to inform the current connection state to respective processors by inserting data indicating the current connection state into a packet to be distributed.

42. (Previously Presented) An apparatus according to claim 41, wherein a load state of a processor is contained in a table.

43. (Previously Presented) An apparatus according to claim 41, wherein a load state of a processor is expressed as a Boolean value.

44. (Previously Presented) An apparatus according to claim 41, wherein a load state of a processor is expressed as value which corresponds to the percentage of load.

45. (Previously Presented) An apparatus according to claim 41, wherein said selection circuitry is configured such that a processor is selected also on the basis of a parameter indicating the service profile supported by a respective processor.

46. (Previously Presented) An apparatus according to claim 45, wherein said parameter is contained in a table.

47. (Previously Presented) An apparatus according to claim 41, further comprising: data insertion circuitry configured to insert a communication connection state into a received packet to be routed.

48. (Previously Presented) An apparatus according to claim 41, wherein the processors are comprised of multicore digital signal processing elements having a shared data storage for all cores, whereby said device comprises a first level of load balancing configured to select a digital signal processing means and a second level of load balancing configured to select a single core.

49. (Previously Presented) An apparatus according to claim 41, further comprising:

a switch configured to redirect excess traffic to another apparatus, wherein said excess traffic is defined upon the number of active processors.

50. (Previously Presented) A system comprising:

an obtaining unit configured to obtain a current connection state as well as a current load state of each of a plurality of processors configured to perform communication in a packet switched connection;

a selector configured to select on a per received packet basis one of said processors, by a load balancer configured to distribute load to said processors in such a manner that a respective next received packet is distributed to the selected processor having a lowest load irrespective of a specific connection to which this next received packet belongs;

a maintenance unit configured to maintain information about the load state of each processor so that said selecting comprises selecting one of said processors to serve and process a respective received packet based on the load states; and

an informing unit configured to inform the current connection state to respective processors comprising inserting data indicating the current connection state into a packet to be distributed.

51. (Previously Presented) A computer program embodied on a computer readable medium, the computer readable medium storing code comprising computer executable instructions configured to perform a method comprising:

obtaining a current connection state as well as a current load state of each of a plurality of processors configured to perform communication in a packet switched connection;

selecting on a per received packet basis, by a load balancer configured to distribute load to said processors, one of said processors in such a manner that a respective next received packet is distributed to said selected one of said processors having a lowest load irrespective of a specific connection to which a respective received packet belongs;

maintaining information about the load state of each processor so that said selecting comprises selecting one of said processors to serve and process a respective received packet based on the load state; and

informing the current connection state to respective processors comprising inserting data indicating the current connection state into a packet to be distributed.

52. (Previously Presented) A system comprising:  
a plurality of processors configured to perform communication in a packet switched connection; and

at least one load balancer configured to distribute the load to said processors,  
wherein the load balancer is configured to

obtain a stored current connection state and a current load state of each of said processors,

maintain information about the load state of each of said processors,

select a processor in such a manner that a respective next received packet is distributed to the processor having a lowest load irrespective of a specific connection to which a respective received packet belongs, and

inform the current connection state to respective processors comprising inserting data indicating the current connection state into a packet to be distributed.

53. (Previously Presented) An apparatus, comprising:

a load balancer, wherein the load balancer is configured to

obtain a current connection state and a current load state of each of a plurality of processors;

maintain information about the load state of each of said processors;

and

select, on a per received packet basis, a processor having a lowest load in such a manner that a respective next received packet is distributed to the processor irrespective of a specific connection to which this next respective received packet belongs.

54. (Previously Presented) An apparatus, comprising:

maintaining means for maintaining a load state of each of multiple processors performing a packet switched communication connection;

selecting means for selecting, on a per received packet basis, one of the processors on the basis of its load state in such a manner that a respective next received packet is distributed to a processor having a lowest load irrespective of a specific connection to which a respective received packet belongs; and

informing means for information the current connection state to respective processors by inserting data indicating the current connection state into a packet to be distributed.

55. (Previously Presented) An apparatus according to claim 54, further comprising: means for inserting a communication connection state into a packet to be routed.

56. (Previously Presented) An apparatus according to claim 54, further comprising: means for redirecting excess traffic to another device, wherein said excess traffic is defined upon the number of active processors.

In re: Koistinen et al.  
Appl. No.: 10/518,930  
Filing Date: 1/13/2005

9. ***Evidence Appendix.***

None.

In re: Koistinen et al.  
Appl. No.: 10/518,930  
Filing Date: 1/13/2005

10. *Related Proceedings Appendix.*

None.

In re: Koistinen et al.  
Appl. No.: 10/518,930  
Filing Date: 1/13/2005

**CONCLUSION**

In light of the remarks presented herein, Applicant submits that Claims 26-37 and 41-56 are patentable and the rejections should be reversed.

Respectfully submitted,



Jonathan A. Thomas  
Registration No. 62,200

**CUSTOMER No. 00826**  
**ALSTON & BIRD LLP**  
Bank of America Plaza  
101 South Tryon Street, Suite 4000  
Charlotte, NC 28280-4000  
Tel Charlotte Office (704) 444-1000  
Fax Charlotte Office (704) 444-1111  
LEGAL02/31552861v1

ELECTRONICALLY FILED USING THE EFS-WEB ELECTRONIC FILING SYSTEM OF THE UNITED STATES PATENT & TRADEMARK OFFICE ON OCTOBER 19, 2009.